sujet de stage de Master 2 recherche (2023)

Types for sensitivity analysis and differential privacy in functional programming

Advisor: Patrick Baillot (patrick.baillot@univ-lille.fr)
Website: `https://pro.univ-lille.fr/patrick-baillot/`

Research lab: CRIStAL, Université de Lille, Équipe *SyCoMoRES*
Lab website: `https://www.cristal.univ-lille.fr/?lang=fr`

Duration: 4-6 months
Keywords: type systems, program sensitivity, differential privacy, linear logic, bunched logic

Context:

Program *sensitivity* bounds the distance between the outputs of a program when run on two related inputs. This notion plays an important role in differential privacy, a rigorous approach for ensuring privacy in database queries and data analysis computation. Several programming languages approaches to sensitivity analysis and differential privacy have been developed in the last decade [BGHP16]. Among them are type systems inspired by linear logic, as introduced in the Fuzz programming language [RP10, GHH+13]. In Fuzz, each type is equipped with its own notion of distance, and sensitivity analysis is carried out by type checking. The language is also equipped with a monadic type for probabilistic computation. This leads to theorems stating that if a program is well-typed in this system, then it is differentially private.

Fuzz was designed to account for two notions of distances on product types: L_1 (or Manhattan) distance and L_∞ (or Chebyshev) distance. This is because these distances play an important role in differential privacy. However, more general L_p distances (such as e.g. euclidean distance L_2) are used in optimisation, information theory and statistics. In [jwdABG22] (joint work with colleagues at Boston University) an extension of Fuzz was proposed, called Bunched Fuzz, with a richer type system allowing to account for arbitrary Lp distances. This system is inspired by the logic of Bunched Implications (BI) [OP99], hence its name: instead of a set structure, typing environments have a bunch (tree) structure.

Objectives:

In this internship we propose to investigate and enlarge the study of properties of Bunched Fuzz. This work could take one or several of the following directions, according to the interests of the candidate:

- Deepen the understanding of the meta-properties of Bunched Fuzz: one can for instance study a subject-reduction property, for a certain operational semantics.
- Extension of typing rules for distances on probability distributions: Bunched Fuzz allows for the definition of new distances on the type of probability distributions (such as Hellinger distance); one can study which distances over distributions (defined by f-divergences) admit valid typing rules, which subtyping rules can be added, and which probabilistic computation can be analysed thanks to these new types.
- Comparison with other systems for differential privacy: The Duet language [NDA+19] can be seen as a refinement of Fuzz allowing to analyse the more general notion of (ϵ, δ)-differential privacy. However it is not equipped with type constructs handling L_p distances. One can wonder if it could be refined in a similar way as Bunched Fuzz. Another setting is that of the Fuzzi system [ZRH+19], a 3-layered logic

(in the sense of program logics) providing a more fine-grained analysis of differential privacy. Could Fuzzi be extended to take into account L_p distances?

Expected background: Some knowledge of type systems, functional programming or λ-calculus is needed. Some familiarity with linear logic or linear type systems would also be useful. Knowledge of differential privacy would be appreciated but is not compulsory.

References

[BGHP16] Gilles Barthe, Marco Gaboardi, Justin Hsu, and Benjamin C. Pierce. Programming language techniques for differential privacy. *ACM SIGLOG News*, 3(1):34–53, 2016.

[GHH+13] Marco Gaboardi, Andreas Haeberlen, Justin Hsu, Arjun Narayan, and Benjamin C. Pierce. Linear dependent types for differential privacy. In *POPL '13*. ACM, 2013.

[jwdABG22] june wunder, Arthur Azevedo de Amorim, Patrick Baillot, and Marco Gaboardi. Bunched Fuzz: Sensitivity for vector metrics. *CoRR*, abs/2202.01901, 2022. to appear in the Proceedings of ESOP 2023 (European Symposium on programming).

[NDA+19] Joseph P. Near, David Darais, Chike Abuah, Tim Stevens, Pranav Gaddamadugu, Lun Wang, Neel Somani, Mu Zhang, Nikhil Sharma, Alex Shan, and Dawn Song. Duet: an expressive higher-order language and linear type system for statically enforcing differential privacy. *Proc. ACM Program. Lang.*, 3(OOPSLA), 2019.

[OP99] Peter W. O'Hearn and David J. Pym. The logic of bunched implications. *Bull. Symb. Log.*, 5(2), 1999.

[RP10] Jason Reed and Benjamin C. Pierce. Distance makes the types grow stronger: a calculus for differential privacy. In *ICFP 2010*. ACM, 2010.

[Za] Hengchu Zhang and Edo Roth and.